Translated by: mohammad madvari

بنام خدا

ترجمه اسلاید های درس مهندسی نرم افزار پرسمن

فصل های ۱ تا ۶

گردآورنده : محمد مدواری

Translated by: mohammad madvari

فصل اول

نرم افزار و مهندسی نرم افزار

نقش های دو گانه نرم افزار

- **نقش اول: نرم افزار یک محصول است.**
  - **دارای پتانسیل محاسباتی است.**
  - **تولید . مدیریت . تغییر . نمایش و انتقال اطلاعات را انجام می دهد.**
- **نقش دوم: نرم افزار وسیله ای برای تحویل محصول است.**
  - **از functionalityسیستم پشتیبانی می کند یا به طور مستقیم آن را فراهم می کند.**
  - **سایر برنامه ها را کنترل می کند (مثل سیستم عامل ها)**
  - **بر روی ارتباطات تاثیر می گذارد (مثل نرم افزارهای شبکه)**
  - **کمک به ساخت سایر برنامه ها می کند (مثل ابزارهای نرم افزاری)**

یک نرم افزار وقتی موفق است که :

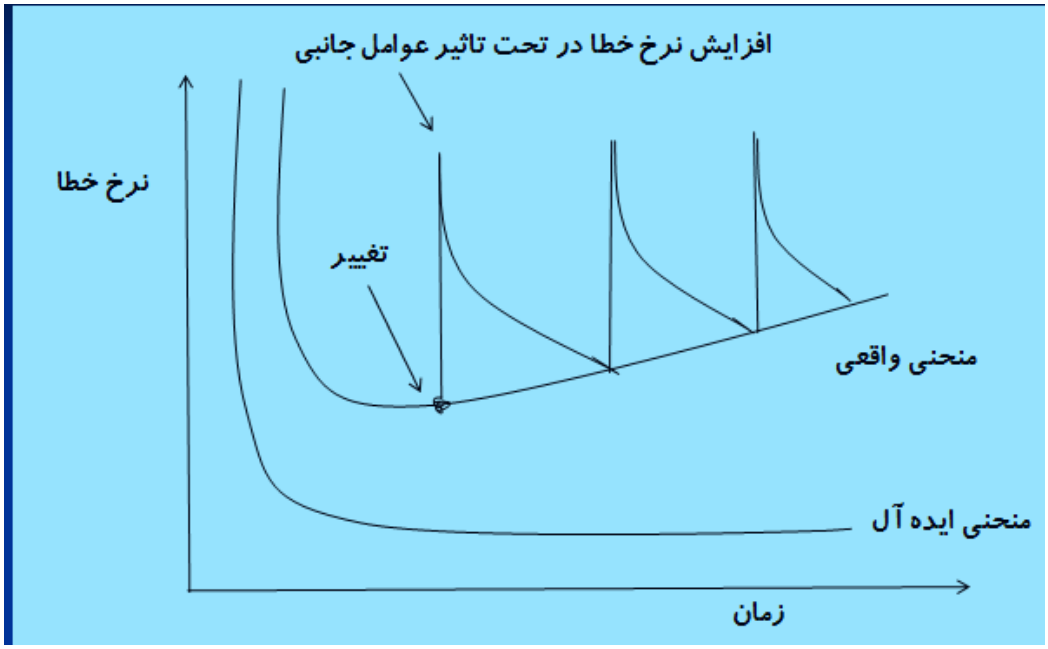
- **نیازهای کاربران را برطرف کند.**
- **بدون مشکل کار کند.**
- **اصلاح و تغییر بر روی آن ساده باشد.**
- **استفاده از آن آسان باشد.**

Translated by: mohammad madvari

نرم افزار چیست؟

- نرم افزار یک واحد منطقی است نه فیزیکی
  - نرم افزار توسعه داده می شود یا مهندسی می شود.
  - نرم افزار رو به زوال می رود ولی کهنه نمی شود.
  - نرم افزار پیچیده است به این دلیل که بیشتر سفارشی ساخته می شود.

نمودار زوال پذیری یک نرم افزار



- مهندسی نرم افزار سعی می کند شیب منحنی واقعی را کاهش دهد.

کاربردهای نرم افزار

- محتوای اطلاعاتی و هدف برنامه ماهیت و کاربرد نرم افزار را مشخص می کنند.
- نرم افزار سیستمی برای سرویس دهی به سایر برنامه ها نوشته می شود.

Translated by: mohammad madvari

- نرم افزار کاربردی نیازهایی خاص مثل نیازهای تجاری را رفع می کند.
- نرم افزارهای مهندسی/علمی مثل نرم افزارهای CAD
- نرم افزارهای جاسازی شده مثل محصولات هوشمند
- نرم افزار های خط تولید
- نرم افزاری وب
- نرم افزارهای هوش مصنوعی

انواع جدید نرم افزار

- نرم افزارهای محاسباتی فراگیر (ubiquitous computing) – مثل نرم افزارهای شبکه های بی سیم
- نرم افزارهای Netsourcing – وب مثل یک موتور محاسبه عمل می کند.
- نرم افزارهای Open source
- همچنین . . .
  - نرم افزارهای Data mining
  - نرم افزارهای Grid computing
  - نرم افزارهای Cognitive machines
  - نرم افزارهایی برای nanotechnologies

نرم افزار میراثی

- نرم افزار باید با محیط ها و تکنولوژی های جدید سازگار شود.
- نرم افزار باید توسعه داده شود تا نیازهای تجاری جدید را برطرف کند.
- نرم افزار باید گسترش پیدا کند تا با سیستم ها و پایگاه های داده ای جدید بتواند کار کند.
- معماری نرم افزار باید به گونه ای دوباره ایجاد شود که در محیط های شبکه ای بتواند کار کند.
- مساله اصلی این است که کیفیت نرم افزار معمولا پایین است.

تکامل نرم افزار

- قانون تغییر مداوم
- قانون افزایش پیچیدگی
- قانون تنظیم خودکار
- قانون حمایت از ثبات سازمانی
- قانون حمایت از آشنایان
- قانون پیشرفت پیوسته
- قانون کاهش کیفیت

Translated by: mohammad madvari

- قانون سیستم بازخوردی

باور های غلط مشتریان :

- همه ی شما به یک دستورالعمل نیاز دارید که به شما بگوید چگونه برنامه نویسی را شروع کنید.
- نیازمندی ها تغییر می کنند ولی نرم افزار می تواند خودش را با انعطافی که دارد با آنها وفق دهد.

باورهای غلطی که توسعه دهنگان نرم افزار می توانند داشته باشند:

- وقتی که برنامه ما اجرا شد کار تمام است.
- تا زمانی که برنامه در حال اجرا است من هیچ نظری در مورد چگونگی عملکرد آن ندارم.
- تنها چیز قابل تحویل برنامه ای است که کار می کند.
- مهندسی نرم افزار باعث ایجاد مستندات غیر ضروری و حجیم می شود که نتیجه ی آن افت سرعت ما است.

باورهای غلطی مدیریتی :

- کتاب استانداردها برای ساخت نرم افزار نیازهای اطلاعاتی توسعه دهندگان را برطرف می کند.
- اگر از روال عادی کار عقب افتادیم می توانیم افراد بیشتری را جذب کنیم.
- می توانیم تولید نرم افزار را به شرکت دیگری بسپاریم و خودمان استراحت کنیم.

فصل ۲

فرایند

Translated by: mohammad madvari

فرایند نرم افزاری

- فرایند نرم افزاری (software process) چیست؟
- فعالیت های چارچوب (Framework activities) چه هستند؟
- فرایند های مدلسازی شده چگونه هستند؟
- طرح های فرایند (Process patterns) چه هستند؟
- ویژگی های مدلهای افزایشی (Incremental) چیست؟
- فرایند یکپارچه شده (Unified) به چه معناست؟
- چرا Agility یک کلمه کلیدی است؟
- توسعه نرم افزاری سریع به چه معناست؟

توسعه ی نرم افزار

- یک فرایند یادگیری اجتماعی است.
  - مکالمه (dialogue)
  - تعامل (interaction)
  - بازگشتی بودن (iterative)
- چه چیزهایی نیاز داریم؟
  - مراحل قابل پیش بینی
  - نتایج به موقع و با کیفیت بالا
- فرایند فراهم می کند:
  - پایداری
  - کنترل
  - سازمان دهی
  - ولی تنها فعالیت هایی مناسب هستند که کنترل و مستندسازی را انجام می دهند.

ارزیابی فرایند

- تعداد مکانیسم ها وابزارهای موجود
- کیفیت . تناسب زمانی و دوام طولانی شاخص های برجسته ای برای سنجش خوب بودن یک فرایند هستند.

Translated by: mohammad madvari

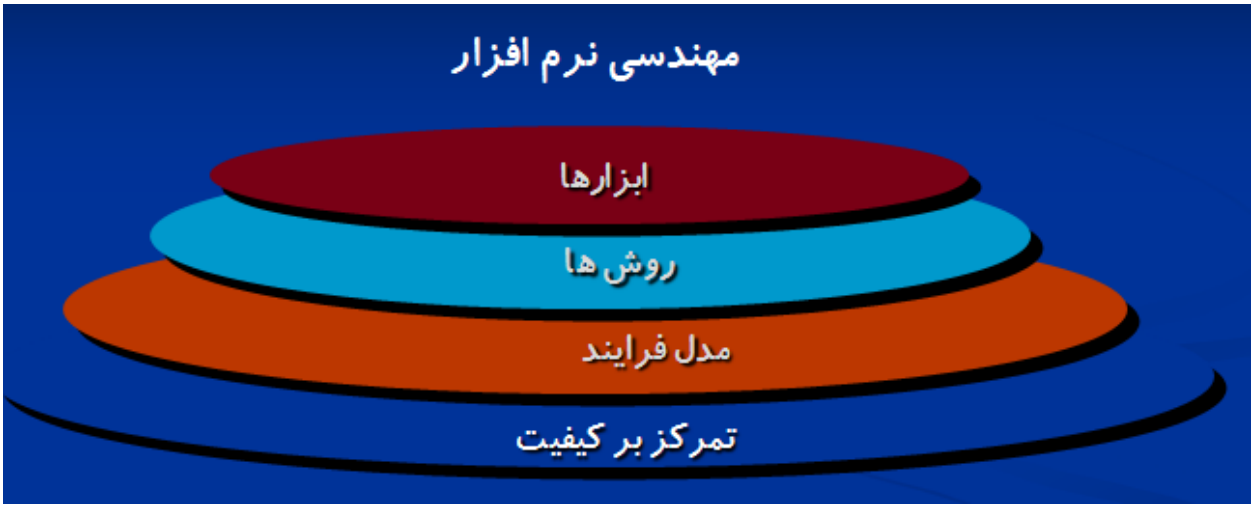
فرایند نرم افزاری

- تعریف: یک چارچوب برای کارهایی که باعث ایجاد یک نرم افزار با کیفیت بالا می شوند.
- رویکرد مورد استفاده را تعریف می کند.
- مهندسی نرم افزار یعنی فرایند نرم افزاری به اضافه ی تکنولوژی هایی که فرایند را ایجاد می کند.

مهندسی نرم افزار

- کاربرد سیستماتیک منظم و رویکرد قابل سنجش برای توسعه . عملکرد و نگه داری نرم افزار.
- مطالعه ی رویکردهای کاربردی فوق الذکر
- مهندسی نرم افزار یک تکنولوژی لایه ای است.

تکنولوژی لایه ای



- مهندسی نرم افزار باید تعهدی برای فراهم آوردن کیفیت داشته باشد.
  - تمرکز کیفیت عنصری پایه ای برای پشتیبانی از مهندسی نرم افزار است.
- مهندسی نرم افزار روش ها و ابزارهای فرایند است.
  - لایه فرایند
    - لایه های تکنولوژی را در کنار هم نگه می دارد.
    - یک چارچوب تعریف می کند.
    - اساس کنترل و مدیریت را تشکیل می دهد.
    - برقرار می کند زمینه ای را برای :
      - روش های تکنیکی که به کار گرفته می شوند.

Translated by: mohammad madvari

- محصولات کاری که تولید می شوند.
- نقاط عطفی (milestone) که باید ایجاد شوند.
- اطمینان از کیفیت
- مدیریت تغییرات

- روش ها(methods) چگونگی انجام کارها را مشخص می کنند.
  - این کارها شامل ارتباط . تحلیل نیازمندی ها . مدل سازی طراحی . ساخت برنامه . تست و پشتیبانی از آن است.
- ابزارها از فرایند ها و روش ها بصورت اتوماتیک و یا نیمه اتوماتیک پشتیبانی میکنند.

چارچوب فرایند

- یک چارچوب فرایند :
  - پایه ای برای فرایند نرم افزاری کامل است.
  - تعدادی از فعالیت های چارچوب را مشخص می کند.
    - قابل اعمال به کلیه پروژه های نرم افزاری است.
    - صرف نظر از میزان پیچیدگی آن
  - شامل مجموعه ای از فعالیت های چتری (Umbrella activities) است.
    - این ها قابل اعمال به کل پروژه های نرم افزاری هستند.

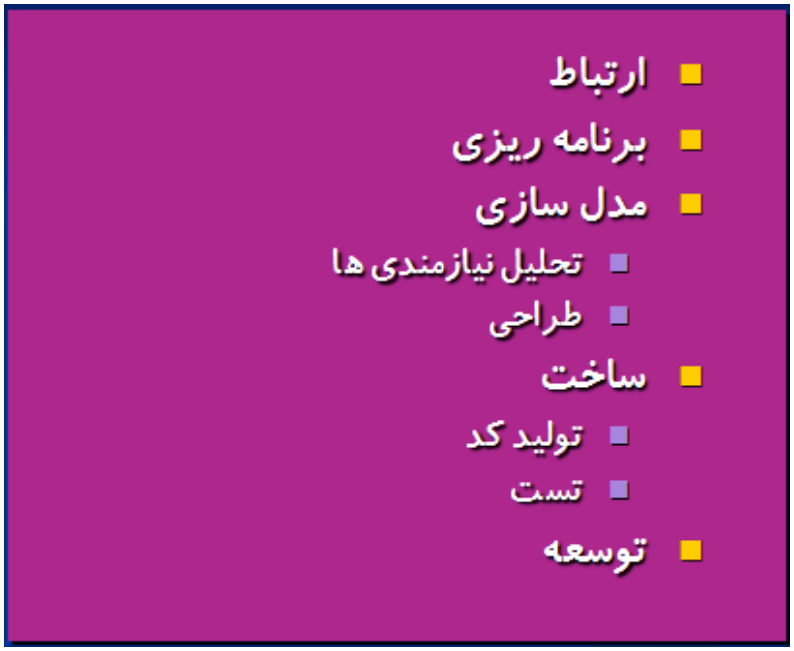
یک چارچوب فرایند

- هر فعالیت چارچوب مجموعه ای از Action های مهندسی نرم افزار را داراست.
  - مجموعه ای از وظائف مرتبط که محصول کاری major (عمده) مهندسی نرم افزار را تولید می کنند.

Translated by: mohammad madvari

- هر عمل (Action) با وظائف کاری ای که بعضی اجزای آن کار انجام می دهند اجرا می شود.

فعالیت های چارچوب



- ارتباط
- برنامه ریزی
- مدل سازی
  - تحلیل نیازمندی ها
  - طراحی
- ساخت
  - تولید کد
  - تست
- توسعه

- ارتباط
  - همکاری با مشتری
- برنامه ریزی
  - طرحی که وظائف کاری قابل انجام . ریسک ها . منابع . محصولات کاری و زمان بندی ها را نشان

Translated by: mohammad madvari

می دهد.

- مدل سازی
  - ایجاد مدلهایی برای فهم بهتر نیازمندی های و ارتباطات نرم افزار
- ساخت
  - تولید کد همراه با تست آن
- توسعه
  - نرم افزار به مشتریانی که آن را ارزیابی کردند و بازخوردی را برای آن فراهم کردند تحویل داده می شود.

مثال

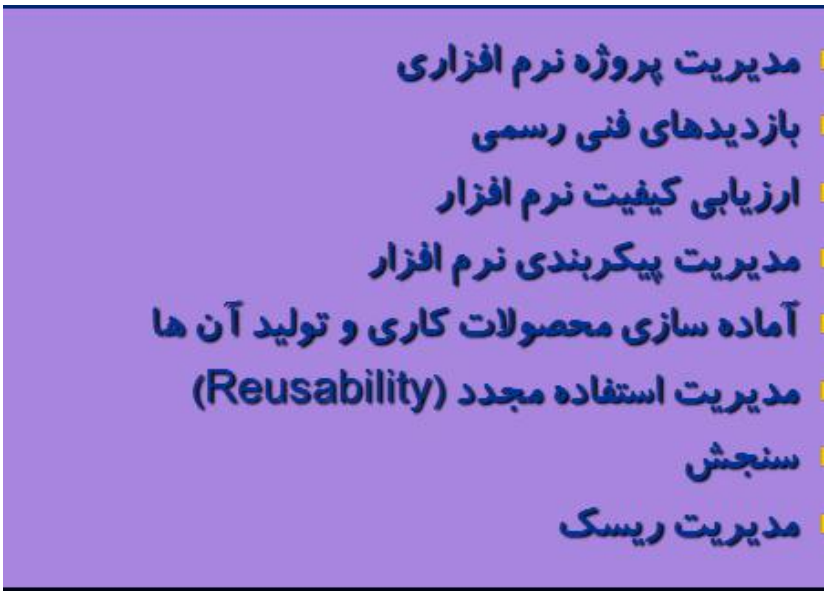
- فعالیت مدل سازی
  - متشکل از ۲ Action مهندسی نرم افزار
    - تحلیل و طراحی
      - وظائف کاری تحلیل عبارتند از :
        - جمع آوری نیازمندی ها
        - جزئیات (پیچیدگی)
        - مذاکره
        - ذکر خصوصیات
        - اعتبار سنجی
      - محصولات کاری تحلیل
        - مدل تحلیل و تشخیص نیازمندی ها
- پروژه های مختلف مجموعه ی وظائف مختلفی نیز دارند. تیم نرم افزاری مجموعه وظائف را بر اساس مسائل و ویژگی های پروژه انتخاب می کنند.

فعالیت های چتری (Umbrella activities)

- چارچوب بوسیله ی تعدادی از فعالیت های چتری کامل می شود.
- فعالیت های چتری در کل فرایند نرم افزاری قرار دارند و تمرکز آن ها بر مدیریت . پیشرفت و کنترل پروژه است.

فعالیت های چتری

Translated by: mohammad madvari



- مدیریت پروژه نرم افزاری
- بازدیدهای فنی رسمی
- ارزیابی کیفیت نرم افزار
- مدیریت پیکربندی نرم افزار
- آماده سازی محصولات کاری و تولید آن ها
- مدیریت استفاده مجدد (Reusability)
- سنجش
- مدیریت ریسک

مدل فرایند : سازگاری

- فعالیت های چارچوب همیشه بر روی هر پروژه ای اعمال می شوند ولی ...
- وظائف (و درجه ی سختی آن ها) برای هر فعالیت بر اساس موارد زیر متفاوت است:
  - نوع پروژه
  - ویژگی های پروژه
  - قضاوت صحیح ; رضایت تیم پروژه

CMMI

- CMMI کارهای مشخصی را که برای رسیدن به هدف باید انجام شوند را تعیین می کند و شامل

Translated by: mohammad madvari

دو قسمت است:

- Specific goals
- *Specific Practices*

آیا ما باید از CMMI استفاده کنیم؟

- **همیشه باید ماهیت CMMI پذیرفته شود.**
- **CMMI می گوید که توسعه نرم افزار باید به طور جدی پیگیری شود.**
  - **برای توسعه نرم افزار باید برنامه ریزی . کنترل . بررسی با دقت و هدایت به صورت حرفه ای صورت گیرد.و تمرکز آن باید بر نیازهای مشتریان و کاربران نهایی و مهارت مهندسین و کیفیت محصول نهایی باشد.**
- **CMMI ممکن است برای تمام سازمان ها و تمام زمان ها مناسب نباشد.**

Process Pattern ها

- **Process pattern مجموعه ای از فعالیت ها . عملیات . وظائف کاری . محصولات کاری و یا رفتارهای مرتبط می باشد.**
- **از یک Template (الگو)برای تعریف یک Pattern استفاده می شود.**
- **مثال هایی در این زمینه :**
  - **ارتباط با مشتری (فعالیت)**
  - **تحلیل (عملیات)**
  - **جمع آوری نیازمندی ها (وظیفه)**
  - **بازبینی محصول کاری (وظیفه)**
  - **مدل طراحی (محصول کاری)**

- **یک Template (الگو) روشی برای توصیف یک ویژگی مهم فرایند نرم افزاری است.**
- **با ترکیب Pattern ها تیم نرم افزاری فرایندی را می سازد که نیازهای مشتری را به بهترین نحو برآورده می کند.**
- **محتویات Template**
  - **نام طرح**
  - **هدف**
  - **نوع**
  - **فضای اولیه**

- مسئله
- راه حل
- فضای نهایی
- Pattern های مرتبط

ارزیابی فرایند

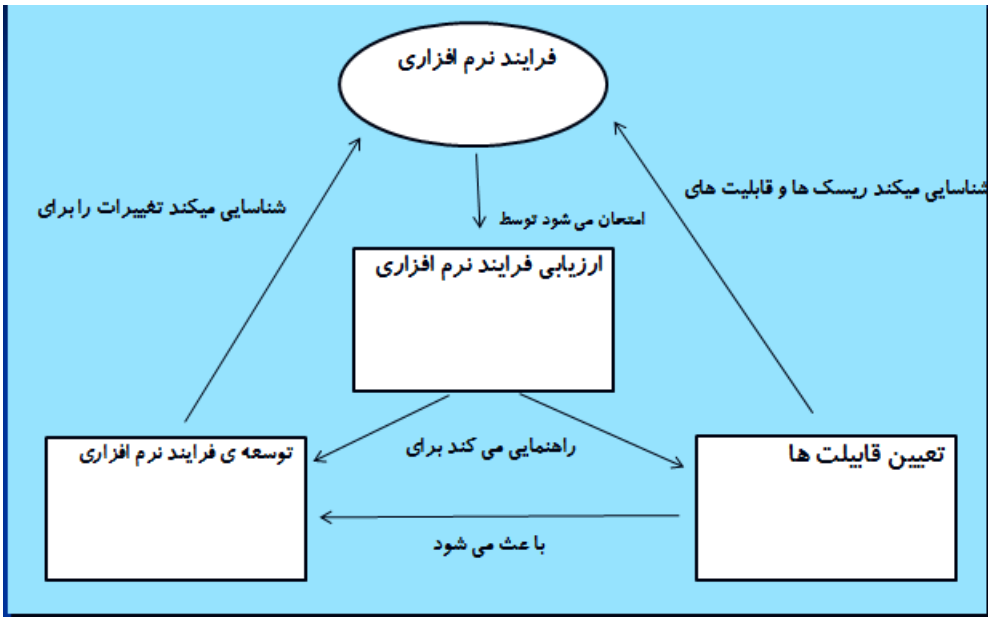
- فرایند تضمین اینکه نرم افزار در زمان مقرر تحویل داده شود ونیازهای مشتری را برآورده کند نمی دهد.
- فرایند به این دلیل ارزیابی می شود که از نحوه ی برخورد آن با بحران های فرایندی که برای مهندسی نرم افزار موفق ضروری هستند مطمئن شویم.
  - یک ایده برای ارزیابی آن بهبود دادن آن است.
- در حال حاضر روش های ارزیابی مختلفی وجود دارد که عبارتند از :
  - SCAMPI
  - CBA IPI
  - SPICE
  - ISO 9001:2000

ISO 9001:2000

- بعنوان یک چرخه ی برنامه ریزی-انجام-تست-عمل (Plan-Do-Check-Act) پذیرفته شده است.
  - Plan اهداف فرایند . فعالیت ها و وظائفی را که برای رسیدن به کیفیت بالای نرم افزار و رضایت مشتری ضروری است را مشخص می کند.
  - Do فرایند های نرم افزاری که شامل فعالیت های چارچوب و چتری هستند را انجام می دهد.
  - Check فرایند را برای اطمینان از اینکه تمام نیازمندی ها برای مدیریت کیفیت انجام شده اند بررسی می کند.
  - Act فعالیت های توسعه ی فرایند نرم افزاری را شروع می کند.

ارزیابی و توسعه

Translated by: mohammad madvari



فرایند نرم افزاری شخصی (PSP)

- ۵ فعالیت چارچوب زیر را توصیه می کند :
  - برنامه ریزی
  - طراحی سطح بالا
  - بازبینی طراحی سطح بالا
  - توسعه
  - Postmortem
- بر نیازهای هر مهندس نرم افزار برای شناسایی سریع خطاها جهت فهم نوع خطا ها تاکید دارد.

فرایند نرم افزاری تیمی (TSP)

- هر پروژه با استفاده از یک Script که وظائف اجرایی را تعریف می کند شروع می شود.
- تیم ها خود گردانند.
- معیار های سنجش تقویت می شوند.
- معیارهای سنجش با هدف توسعه ی فرایند تیمی تحلیل می شوند.

Translated by: mohammad madvari

- هر دوی PSP و TSP به کار سخت . آموزش و هماهنگی نیاز دارند.
- سعی هر دو بر این است که محصول تولیدی به سمت کیفیت و منافع بیشتر برود.
- در هر دو حالت اگر فرایند انتخاب شده ضعیف باشد نتیجه ی سوئی در محصول نهایی خواهد داشت.

هدف اصلی هر فرایند نرم افزاری: کیفیت بالا

یادآوری:

کیفیت بالا = انجام به موقع پروژه

چرا ؟

چون دوباره کاری کمتر است.

فصل ۳

مدل های فرایندی

Prescriptive(نسخه ای)

مدل های Prescriptive

- این مدل ها از یک شیوه ی ترتیبی برای مهندسی نرم افزار استفاده می کنند.

Translated by: mohammad madvari

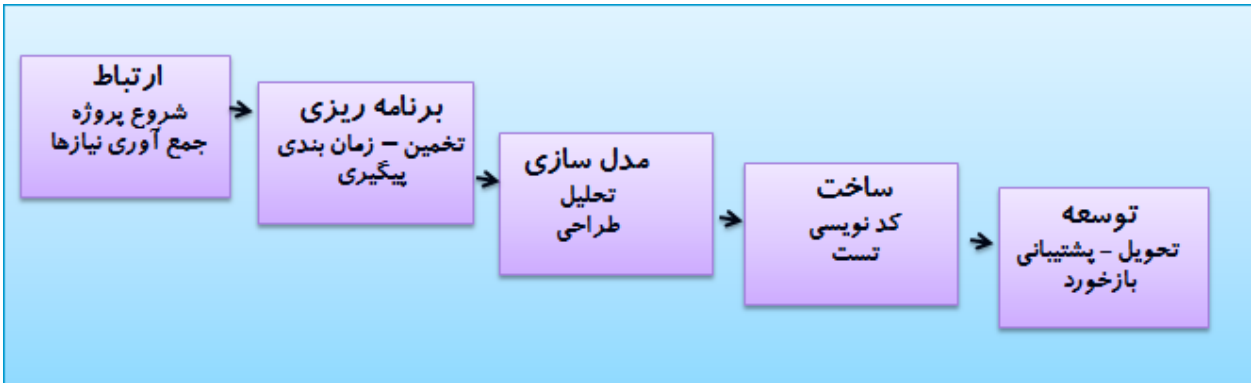
سوالاتی که دراین زمینه پیش می آید ...

- اگر مدل های prescriptive برای ایجاد یک ساختار و ترتیب تلاش کنند آیا برای نرم افزارهایی که با تغییرات پیشرفت می کنند مناسب نیستند ؟
- در عین حال اگر ما مدل های فرایندی سنتی را کنار بگذاریم و آن ها را با مدل های کم ساخت یافته تر جایگزین کنیم آیا ما را قادر به رسیدن به هماهنگی و وابستگی در کار نرم افزاری می سازند؟

- **مدل های فرایندی** Prescriptive
  - یک مجموعه ی مشخص از فعالیت ها . عملیات . وظائف . Milestone ها و محصولات کاری را تعریف می کند.
  - سازگاری . کنترل و سازمان دهی را فراهم می کند.
  - تیم نرم افزاری را از طریق مجموعه ای از فعالیت های چارچوب که بصورت یک نمودار فرایند که می تواند بصورت خطی . افزایشی یا تکاملی باشد راهنمایی می کند.
  - هدفش ایجاد ساختار و نظم و ترتیب است.

- مدل های زیر Prescriptive نامیده می شوند بخاطر اینکه حالت نسخه ای دارند.
  - یک مجموعه از عناصر فرایند
    - فعالیت های چارچوب . عملیات مهندسی نرم افزار . وظائف . محصولات کاری . مکانیسم های ارزیابی کیفیت . مکانیسم های کنترل تغییر برای هر پروژه
  - یک نمودار گردش کار

مدل آبشاری



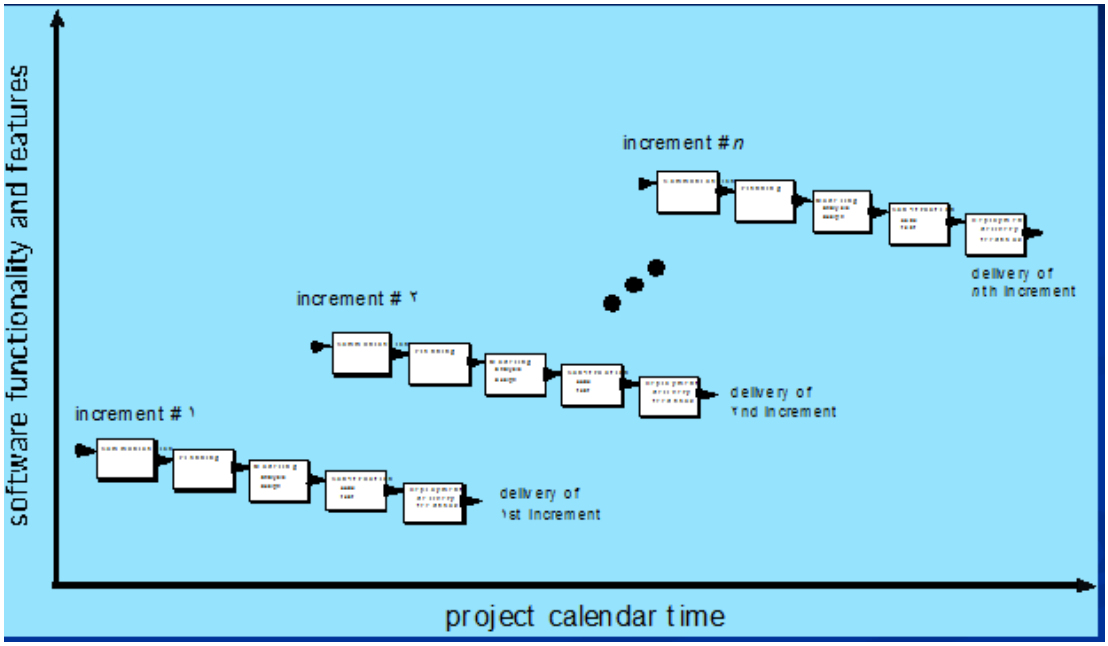
- وقتی از این مدل استفاده می شود که نیازمندی ها قابل فهم و پایدار باشند.

Translated by: mohammad madvari

- سازگاری ها خوب تعریف شده باشد.
- سیستم موجود باشد.

- **چرا این مدل بعضی اوقات شکست می خورد؟**
  - پروژه های واقعی به ندرت از یک نمودار ترتیبی تبعیت می کنند.
  - برای مشتری توضیح تمام نیازمندی ها به طور شفاف سخت است.
  - نسخه ی آزمایشی خیلی دیر آماده می شود.
- **مدل آبشاری برای جایی مناسب است که محدودیت زمانی نداشته باشیم و تغییرات هم رخ ندهد.**

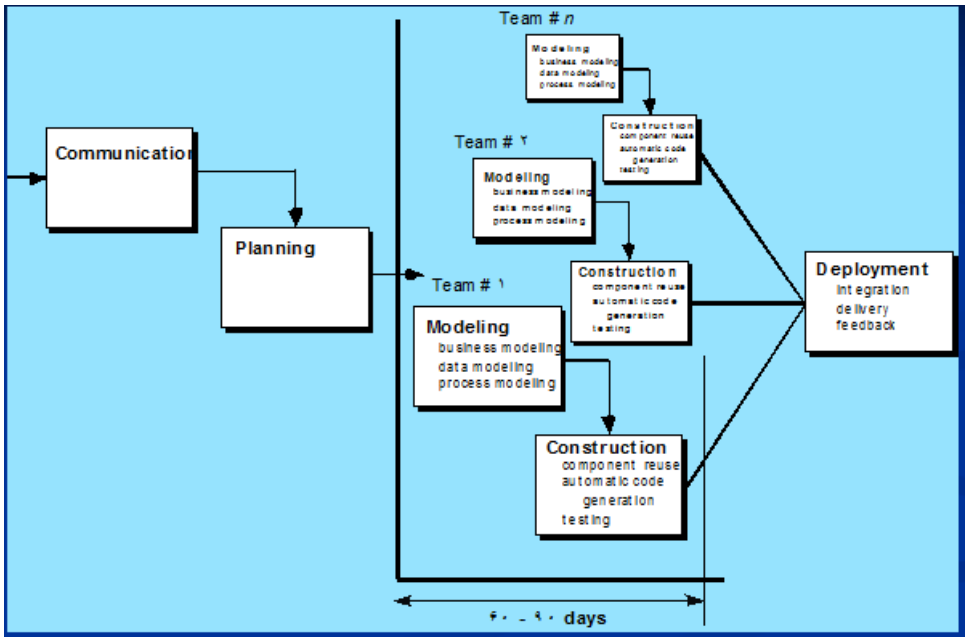
مدل افزایشی



- **عناصر مدل آبشاری را در یک حالت بازگشتی ترکیب می کند.**
- **مجموعه ای از عملکردها را در قالب یک Increment به مشتری تحویل می دهد.**
- **مدل افزایشی بر تحویل یک محصول قابل استفاده در هر Increment است.**
- **اگر مشتری تقاضای دریافت یک increment را در یک تاریخ معین داشت شما می توانید به جای یک increment چندین increment را به او تحویل دهید تا در مراحل بعد بتوانید به استراحت بپردازید.**

مدل RAD

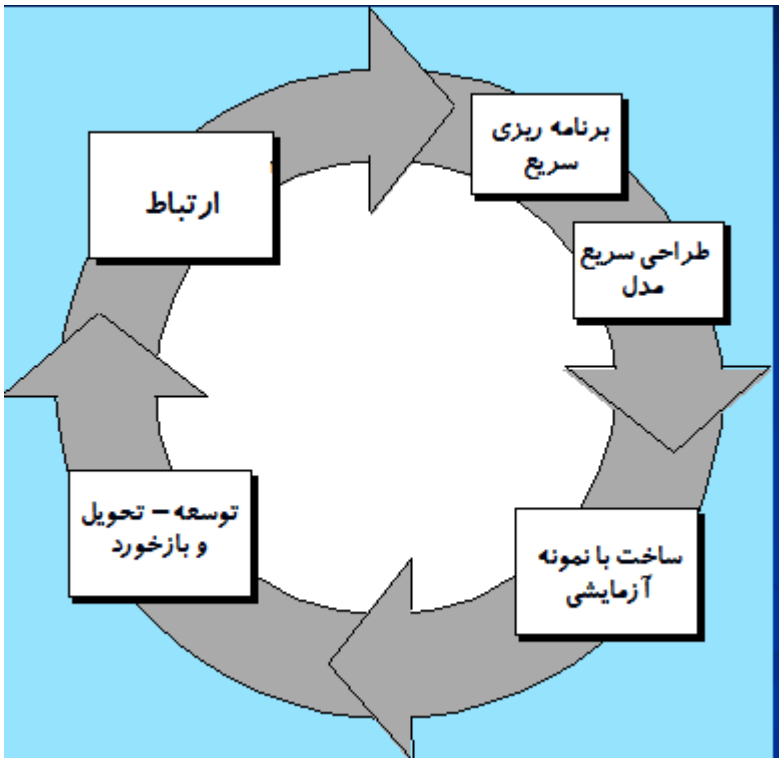
Translated by: mohammad madvari



- تاکید مدل Incremental بر چرخه های توسعه ی کوتاه است.
- RAD یک نمونه ی با سازگاری سرعت بالا از مدل آبشاری است.
  - با استفاده از روش ساخت کامپوننت گرا انجام می شود.
- نیازمندی ها باید خوب فهمیده شوند.
- دامنه ی پروژه باید محدود شود.
- هر Function توسط تیم RAD جداگانه ای به صورت موازی با تیم های دیگر درست می شود و آنگاه همه ی تیم ها به صورت یکپارچه محصول نهایی را تولید می کنند.
- عیب ها
  - نیاز به منابع انسانی کافی
  - توسعه دهندگان و مشتریان باید متعهد به ارتباط دو طرفه باشند.
  - سیستم باید ماژولار (قسمت قسمت) باشد.
  - اگر کارایی یک موضوع و مساله ی مهم باشد ممکن است پروژه شکست بخورد.
  - استفاده از تکنولوژی های جدید ممکن است پروژه را با مشکل مواجه کند.

Translated by: mohammad madvari

مدل های تکاملی : نمونه آزمایشی(Prototyping)



- نرم افزار سر وقت تحویل داده می شود.
- تغییر نیازمندی ها باعث تولید محصول جدیدی می شود.
- بعضی اوقات وضعیت بازار اجازه ی تحویل محصول کامل را به ما نمی دهد.
- نیازمندی های اصلی خوب شناخته شده اند و لی جزئیات هنوز باید شناخته شوند.

Translated by: mohammad madvari

فصل ۴

توسعه سریع نرم افزار (Agile)

شعار توسعه سریع نرم افزار

- توسعه سریع(Agile Development) به صورت یک حرکت و نهضت در آمده است.
  - بخاطر غلبه بر نقاط ضعف واقعی مهندسی نرم افزار توسعه داده شده است.
  - مزایای قابل توجهی را فراهم می کند.
  - قابل اعمال به کلیه پروژه ها و افراد و موقعیت ها نیست.
  - رضایت مشتری را بدنبال دارد.
  - تحویل سریع incrementها (تحویل ها در هر مرحله)
  - تشویق تیم های نرم افزاری برای خود سازمان دهی کردن تیم
  - روش های غیر رسمی
  - به حداقل رسانی محصولات کاری (work products)
  - تاکید بر تحویل محصول بر مبنای تحلیل و طراحی
  - تاکید بر ارتباط پیوسته و فعال بین مشتریان و توسعه دهندگان نرم افزار
- اصول Agile می تواند به عنوان یک فلسفه مهم به کلیه کارهای نرم افزاری اعمال شود.
  - اغلب اوقات قابل پیش بینی نیست که یک محصول چگونه سر وقت به بازار تحویل داده شود.
  - گاهی اوقات قادر به تعریف کامل نیازمندی ها قبل از شروع یک پروژه نیست.
  - مهندسین نرم افزار باید به اندازه کافی سریع کار کنند تا بتوانند به تغییرات محیط کاری پاسخ دهند.

Agility چیست؟

- پاسخ موثر (قابل تطبیق و سریع) به تغییرات
- ارتباط موثر بین تمام ذی نفع ها (کاربران–مشتریان و ...)
- ترسیم مشتری در درون تیم
- سازمان دهی یک تیم همچنان که مشغول به کار است.

نتیجه گیری . . .

Translated by: mohammad madvari

- **تحویل سریع increment های نرم افزار**

- **کمیته Agile ۱۲ اصل را برای کسانی که می خواهند به Agility برسند تعریف کرده است:**
  - **اولویت اول:راضی نگه داشتن مشتری با تحویل سریع و پیوسته نرم افزارهای ارزشمند**
  - **خوشامد گویی به نیازمندی های تغییر یافته- آنها مزیت های رقابت برانگیز مشتریان هستند.**
  - **Incrementهای کوتاه**
  - **مشتری و توسعه دهندگان باید به طور روزانه با هم کار کنند.**
  - **ایجاد پروژه با تشویق افراد**
  - **مکالمه رو در رو موثرترین راه است.**
  - **کارکرد نرم افزار تعیین کننده ی اصلی میزان پیشرفت است.**
  - **توسعه قابل تحمل – سرعت پایدار**
  - **توجه به مزیت های تکنیکی و طراحی خوب Agility را افزایش می دهد.**
  - **سادگی – حداکثر میزان کار نباید انجام شود.**
  - **تیم های خود سازمان یافته بهترین معماری و نیازمندی و طراحی را ایجاد می کنند.**
  - **تیم نرم افزاری مرتبا کارایی خود را بازبینی می کند و به تبع آن خود را منظم می کند.**

یک فرایند Agile

- **توسط توصیفات مشتری از نیازمندی ها (سناریو) حرکت می کند.**
- **طرح های با مدت حیات کوتاه را تشخیص می دهد.**
- **نرم افزار را به صورت تکراری با تاکید زیاد بر فعالیت های ایجاد کننده توسعه می دهد.**
- **Increment های نرم افزاری چندگانه را تحویل می دهد.**
- **تغییرات رخ داده در در هر increment را جمع و جور می کند.**

حقیقت این است:

- **پیش بینی اینکه کدامیک از نیازمندی ها ثابت و کدامیک قابل تغییرند سخت است.**
- **پیش بینی اینکه قبل از ایجاد نرم افزار چقدر طراحی ضروری است تا اینکه نرم افزار بخواهد طراحی را نشان دهد سخت است.**
- **تحلیل . طراحی . ساخت و امتحان آن چنان که ما می خواهیم قابل پیش بینی نیستند.**

فاکتورهای انسانی

- **چه ویژگی هایی باید در بین مردم برای تشکیل تیم نرم افزاری وجود داشته باشد.**

Translated by: mohammad madvari

- شایستگی
- تمرکز عمومی
- همکاری
- تصمیم گیری – ایجاد قابلیت
- مشکل عدم قطعیت – قابلیت حل
  - مسئله ای که امروز باید حل بشه نه فردا
- اعتماد و احترام متقابل
- خودسازماندهی بودن
  - یک تیم خود سازمان دهی شده تمام کارهای انجام شده را در کنترل دارد. تیم تعهدات و اهداف خود را مشخص می کند و طرح هایی را برای رسیدن به آنها تصویب می کند.
  - هیچ چیز بیشتر از پذیرش مسئولیت و انجام کارها باعث تحرک تیم نمی شود.

مدل های فرایندی Agile

- برنامه نویسی مفرط (XP)
- توسعه نرم افزار انطباقی (ASD)
- روش توسعه سیستم های پویا (DSDM)
- روش Scrum
- روش Crystal
- روش Feature Driven Development (FDD)
- روش Agile Modeling (AM)

برنامه نویسی مفرط (XP)

- استفاده گسترده از فرایندهای Agile توسط کنت بک پیشنهاد شد.
  - از یک رویکرد شی گرا استفاده می کند.
  - مجموعه قوانین و اصولی که مرتبط با ۴ زمینه ی فعالیتی زیر می باشند: طرح ریزی – طراحی – کد زنی – تست

Translated by: mohammad madvari

برنامه ریزی XP

- تیم Agile هر گزارشی را بررسی می کند و بهایی را به آن اختصاص می دهد.
  - گزارش کاربر ویژگی ها و عملکردهای مورد نیاز را شرح می دهد.
- گزارشات دسته بندی می شوند تا به یک increment قابل تحویل مبدل شوند.
- برای تحویل تعهد داده می شود.
  - موافقت بر سر تاریخ تحویل و سایر مباحث پروژه
- بعد از اولین increment با نگاه به سرعت اجرای آن پروژه می توان برای تحویل های بعدی نیز تاریخی را مشخص نمود.
  - اساسا این تعداد گزارشات انجام شده ی مشتری است.
  - برای تخمین تاریخ تحویل و زمان بندی ها استفاده می شود.
  - برای پاسخگویی به انجام یک تعهد استفاده می شود.
- با ایجاد گزارشات کاربر شروع می شود.
  - مشتری می تواند گزارشی را اضافه کند یا ارزش گزارش های موجود را تغییر دهد و یا گزارشات را کنار بگذارد و یا حذفشان کند.

طراحی XP

- از اصول KIS استفاده می کند.
  - طراحی عملکرد های (Functionality)اضافی خسته کننده است.
- تشویق به استفاده از کارت های CRC(رجوع شود به فصل ۸)
  - کارت های CRC برای مشخص کردن و سازماندهی کلاس های شی گرایی مربوط به increment جاری استفاده می شوند.
- برای مسائل پیچیده پیشنهاد می شود از راه حل Spike استفاده شود.(یک نمونه اولیه برای طراحی)
- تقویت Refactoring- یک پالایش تکراری از طرح داخلی برنامه
  - تغییری بر روی رفتار خارجی کد شما ایجاد نمی کند.
  - Refactoring به معنی مواجهه ی پیوسته ی طراحی با سیستم ایجاد شده است.

کد زنی XP

- توصیه به ایجاد یک Unit test برای هر گزارش قبل از شروع کد زنی می کند.
  - هنگامی که کد کامل شد خود می تواند یک unit test باشد.
- تشویق به استفاده از Pair Programming
  - حل مساله بلادرنگ
  - متمرکز کردن توسعه دهندگان

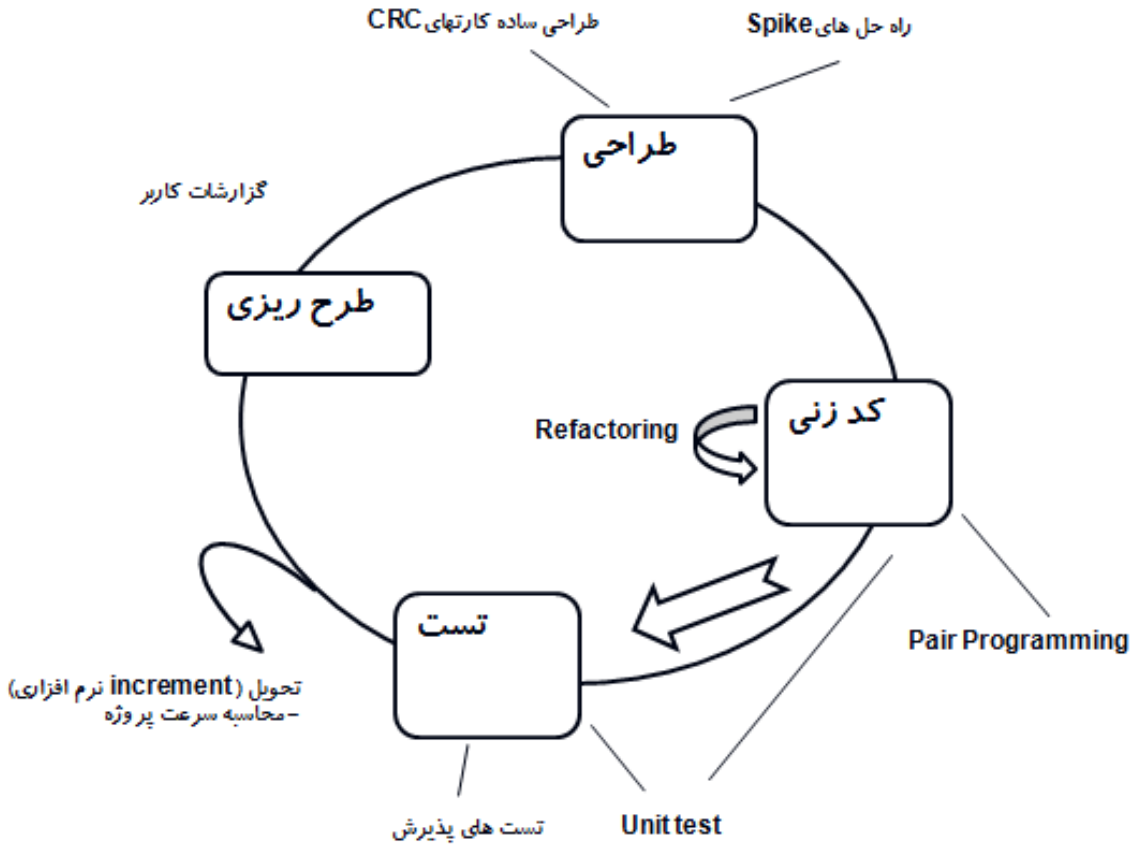
Translated by: mohammad madvari

- همان گونه که کد توسعه داده میشه یکپارچه هم میشه.
  - یکپارچگی کد کمک می کند تا از بروز مشکلاتی نظیرعدم سازگاری نرم افزار و واسط های کاربری نا مناسب اجتناب کنیم و یک محیط Smoke testing را فراهم می کند.

تست XP

- Unit test ها باید با استفاده از یک چارچوب ساخته شوند تا بتوانند به صورت خودکار کار کنند.
  - تاکید به انجام تست پسرفت دارد وقتی که کد تغییر می کند.
- وقتی که تست های فردی سازمان دهی شوند و به صورت رشته ای از تست های عمومی درآیند تست یکپارچه سازی و اعتبارسنجی سیستم می تواند یک روزه انجام شود.
- تست های پذیرش XP (Acceptance tests)بوسیله مشتری تعریف می شوند و برای ارزیابی کل ویژگی های سیستم و عملکردهایی که قابل رویت وبازبینی هستند توسط مشتری اجرا می شوند.

برنامه نویسی مفرط (XP)

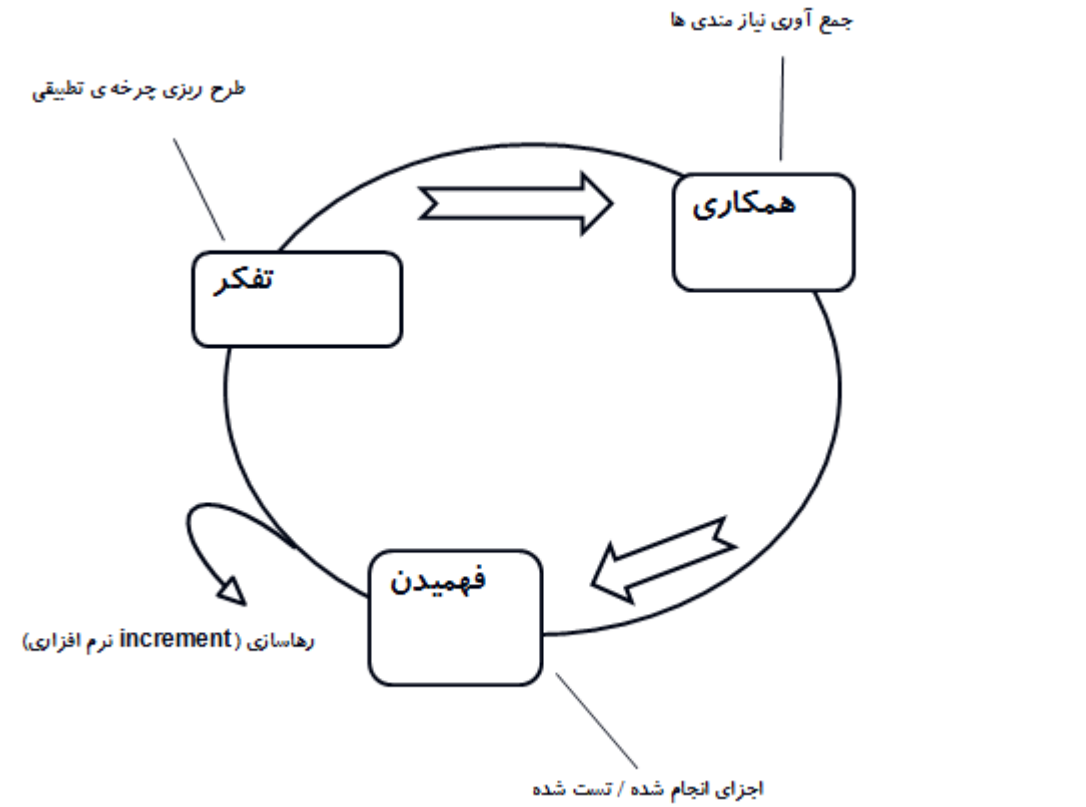


Translated by: mohammad madvari

توسعه نرم افزار انطباقی (ASD)

- تکنیکی برای ایجاد سیستم ها و نرم افزار های پیچیده
- تمرکز آن بر روی همکاری افراد و خود سازمان دهی تیمی است.
- ASD – ویژگی های متمایز
  - طرح ریزی Mission-Driven
  - تمرکز بر روی اجزا (جزء گرایی)
  - استفاده از Time-Boxing (رجوع شود به فصل ۲۴)
  - تصور واضح از ریسک ها
  - تاکید بر همکاری برای جمع آوری نیازمندی ها
  - تاکید بر فهمیدن کلیه ی فرایند ها

توسعه نرم افزار تطبیقی (ASD)



Translated by: mohammad madvari

روش توسعه سیستم های پویا

- چارچوبی برای ایجاد و نگهداری سیستم هایی که از طریق increment های نمونه ای و در یک محیط کنترل شده با محدودیت زمانی مواجه هستند.
- ویژگی های متمایز – DSDM
  - Pareto ی اصلاح شده: ۸۰ درصد پروژه می تواند در ۲۰ درصد زمان انجام شود و در غیر اینصورت به increment بعدی محول می شود.
  - اصول راهنمای ۹ گانه
    - گرفتاری کاربران فعال امری ضروری است.
    - تیم های DSDM باید در تصمیم گیری ها قدرت داشته باشند.
    - تمرکز بر تحویل زود به زود محصول است.
    - معیاری ضروری برای پذیرش محصولات قابل تحویل برای اهداف تجاری است.
    - توسعه ی افزایشی و تکراری برای رسیدن به یک راه حل تجاری صحیح ضروری است.
    - تمامی تغییرات در مراحل توسعه ی نرم افزار برگشت پذیر هستند.
    - نیازمندی ها در یک سطح بالا قابل انجام هستند.
    - عملیات تست در کل چرخه ی حیات تکمیل می شود.

روش Scrum

- Scrum – فعالیتی است که در طی بازی راگبی رخ می دهد.
- ویژگی های متمایز – Scrum
  - کار توسعه در قالب Packet ها تقسیم بندی می شود.
  - تست و مستند سازی تا زمانی که محصول در حال ساخت است انجام می شود.
  - رخدادهای کاری در یک Sprint و مشتق شده از یک Backlog در نیازمندی های موجود
  - ملاقات ها خیلی کوتاه هستند و بعضی اوقات بدون میزگرد انجام می شوند.
  - دموها (Demos) با time boxهای تخصیص داده شده قابل تحویل به مشتری هستند.
- Scrumمجموعه ای از Process Pattern ها و واحد های کاری قسمت بندی شده و بازخورد مشتری را یکپارچه می کند.

روش Crystal

- هدف اصلی آن تحویل مفید و کارکرد خوب نرم افزار و هدف ثانویه آن آمادگی برای تحویل های بعدی است.

Translated by: mohammad madvari

- **ویژگی های متمایز** – Crystal
  - **در واقع خانواده ای از** process model **هاست که اجازه ی** maneuverability **را بر مبنای ویژگی های مسئله می دهد.**
  - **تاکید بر مکالمه ی رودررو دارد.**
  - **اشاره به استفاده از** Reflection workshop **ها برای بازبینی رفتارهای کاری تیم دارد.**

Feature Driven Development

- Process model **های کاربردی برای مهندسی نرم افزار شی گرا**
- **قابل اعمال به پروژه های کوچک بزرگ**
- **ویژگی های متمایز**
  - **تاکید بر تعریف** Feature **ها دارد.**
    - Feature **یک عملکرد کاربر-ارزشی است که در دوهفته یا کمتر قابل انجام است.**
  - **از یک الگوی** feature **استفاده می کند.**
    - <action> the <result> <by | for | of | to> a(n) <object>
  - **یک** feature list **ایجاد میشه و سپس** plan by feature **انجام میشه.**
  - **طراحی و ساخت با هم ادغام می شوند.**
- FDD**تاکید بیشتری بر رهنمود های مدیریت پروژه و تکنیک های آن نسبت به سایر روش های** Agile **دارد.**
- **۶** milestone **را در طی مراحل طراحی و اجرای یک** feature **تعریف می کند که عبارتند از:**
  - **جلسه ی طراحی – طراحی – بازبینی طراحی – کدزنی – بازبینی کد زنی – توسعه ی آن برای ساخت**

مدل سازی سریع

- **قابل اعمال به سیستم های بحرانی تجاری و بزرگ**
- **مسائل مربوط به مدل ها عبارتند از:**
  - **سنگینی – نمادسازی حجیم – درجه ی** formalism **– سختی نگه داری**
- **مجموعه ای از اصول مدلسازی سریع را پیشنهاد می کند:**
  - **مدل با یک هدف همراه باشد.**
    - **چیزهایی که شما با آنها و آنها با شما ارتباط دارند.**
  - **استفاده از مدل های چندگانه**
    - **مدلی را که به ملاقات ها ارزش قائل است انتخاب کنید.**
  - Travel light
    - **هر محصول کاری باید در برابر تغییرات نگه داری شود.**
  - **محتوا مهم تر از ارائه است.**
    - **حتی اگر قرار باشد شما نمادسازی را بخاطر روشن تر شدن مفهوم نقض کنید.**
  - **مدل ها و ابزارهایی که استفاده می کنید را بشناسید.**
  - Adapt locally **(تطبیق دادن محلی = با محل مورد استفاده وفق داده شود)**

Translated by: mohammad madvari

خلاصه

- فلسفه ی agile برای اهمیت مهندسی نرم افزار
  - تیم های خود سازمان یافته
  - ارتباط و همکاری
  - قابلیت تغییر دادن نحوه ی ارائه ی یک فرصت
  - تاکید بر تحویل سریع
- XP
  - استفاده گسترده از فرایند های Agile
  - فعالیت های چارچوب
    - طرح ریزی - طراحی - کدزنی- تست

- روش ASD
  - تاکید بر همکاری تیمی و تیم های خود سازمان یافته دارد.
- روش DSDM
  - سه چرخه تکراری متفاوت را تعریف می کند:
    - مدل های عملکردی
    - طراحی و ساخت
    - پیاده سازی
  - بوسیله فعالیت های چرخه ای امکان سنجی و مطالعه تجاری انجام می شود.
- SCRUM
  - تاکید بر استفاده از مجموعه ای از process pattern ها دارد.
- Crystal
  - خانواده ی مدل فرایندی Agile که می تواند بعنوان ویژگی مشخص یک پروژه پذیرفته شود.
- Feature Driven Development
  - قدری نسبت به بقیه رسمی تر است.
  - نگه داری Agility با تمرکز تیم بر توسعه ی featureها
- Agile Modeling
  - معمولا مدل سزی برای تمام سیستم ها ضروری است ولی بایستی پیچیدگی و نوع و اندازه ی مدل با شروع پروسه ی ایجاد نرم افزار هماهنگ شود.

Translated by: mohammad madvari

فصل ۵

Practice

Practice چیست؟

- **مجموعه ی مفاهیم. اصول .روش ها و ابزارهایی که برای برنامه ریزی و تولید نرم افزار نیاز داریم.**
- Practice **جزئیات .موارد فنی و راه حل هایی را ارائه می دهد که شما برای تولید یک نرم افزار با کیفیت بالا به آن نیاز خواهید داشت.**
- **عناصر یک** Practice **صرف نظر از مدل فرایندی که انتخاب می شود عبارتند از:**
  - **مفاهیم** (Concepts)
  - **اصول** (Principles)
  - **روش ها** (Methods)
  - **ابزارها** (Tools) **از کارکرد روش ها** (Methods) **پشتیبانی می کنند.**

ماهیت Practice

- **جورج پولیا در کتاب خود ماهیت** Practice **را این گونه توصیف می کند:**
  - ***فهم مساله*** *(Communication and analysis)*
    - **ذی نفع ها (مشتری ها و کاربران و...) چه کسانی هستند؟**
    - **چه چیزهایی مجهول هستند؟**
    - **آیا مسئله می تواند به قسمت های مجزایی تقسیم شود؟**
    - **آیا مسئله میتواند به صورت گرافیکی نمایش داده شود؟**
  - ***پیدا کردن یک راه حل*** *(Modeling and software design)*
    - **آیا قبلا مسائلی مشابه این داشتید؟**
    - **آیا مساله مشابه حل شده است؟**
    - **آیا زیرمساله ها می توانند تعریف شوند؟**
    - **آیا شما می توانید راه حلی را ارائه بدهید که در حل بهتر مساله موثر باشد؟**
  - ***اجرا کردن راه حل*** *(Code generation)*
    - **آیا راه حل با برنامه ریزی ها همخوانی دارد؟**
    - **آیا هر جزء از اجزای راه حل درست هستند؟**
  - ***تست نتایج بخاطر اطمینان از صحت آن*** *(*testing and quality assurance*)*
    - **آیا امکان تست هر جزء از اجزای راه حل و جود دارد؟**
    - **آیا نتایجی را که راه حل تولید می کند با عملکرد و ویژگی ها و رفتارهایی که از داده ها مورد انتظار است مطابقت دارد؟**
- Practice **ای خوب است که از روش حل مساله بدیهی استفاده کند.**

Translated by: mohammad madvari

اصول هسته مهندسی نرم افزار

- قبل از شروع یک پروژه ی نرم افزاری مطئمن شوید که نرم افزار یک هدف تجاری دارد و کاربران نیز برای آن ارزش قائلند.
- همه چیز را ساده فرض کن . احمق (قاعده ی KISS)
  - آسان پنداشتن فهم و نگه داری از آن
- همیشه حواست به دیدگاه پروژه و محصول باشد.
  - هیچ وقت طراحی های ناسازگار را به هم وصله دوزی نکن.
- آنچه که شما تولید می کنید دیگران مصرف می کنند.
  - کد برای کسانی که پروژه را نگه داری می کنند مهم است نه برای مشتریان.
- برای آینده آمادگی داشته باش.
  - هیچ وقت بودن هدف یه گوشه ای نشین.
- همیشه ماژول هایت را طوری طراحی کن که قابلیت استفاده مجدد (Reuse) داشته باشند.
  - قابلیت استفاده مجدد (Reuse) از سخت ترین اهداف و کارهاست.
  - از تکنولوژی های شی گرا (OO)استفاده کن.
  - بازدهی سرمایه بستگی به عملکرد تو دارد.
  - مطالعات نشان می دهد که طراحی و ساخت اجزای Reusable بین ۲۵ تا ۲۰۰ درصد برای پروژه هزینه بردار است.
- فکر کن !
  - قبل از انجام هر عملی کاملا در مورد آن فکر کن.

Practice های مهندسی نرم افزار

- **چارچوب کلی فرایند**
  - ارتباط
  - برنامه ریزی
  - مدل سازی
  - ساخت
  - توسعه
- **در اینجا ما شناسایی می کنیم:**
  - اصول اساسی
  - چگونه Practice را شروع کنیم؟
  - یک مجموعه کاری مختصر

Translated by: mohammad madvari

Practice های ارتباطی

- اصول
  - گوش کن
  - قبل از گفتگو خودت را آماده کن.
  - بعضی افراد باید مقدمات گفتگو را فراهم کنند.
  - اگر گفتگو بصورت رو در رو باشد خیلی بهتر است.
  - نتیجه تصمیماتتون را ثبت کنین.
  - با مشتری همکاری داشته باشید.
  - با دقت بر روی تصمیماتتون تمرکز کنین و بحث رو بی نتیجه رها نکنین.
  - اگه یه سری چیزها نامفهوم است از تصاویر استفاده کنین.
  - بحث را در صورتی ادامه بده که...
    - با چیزی موافقت کردید.
    - نتوانستید با چیزی موافقت کنید.
    - یه چیزی مبهم است و الان نمیشه اون را باز کرد.
  - مذاکره وقتی به نتیجه خوبی می رسد که هر دو طرف راضی باشند.
    - مذاکره به معنی موافقت است.

Practice های ارتباطی

- شروع
  - افراد باید از نظر فیزیکی نزدیک هم باشند.
  - ارتباط باید تعاملی و دوطرفه باشد.
  - یک تیم کامل تشکیل داده شود.
  - از ساختار تیمی درستی استفاده شود.
- یک مجموعه کاری خلاصه
  - مشخص کردن افرادی که با آنها نیاز به صحبت است.
  - پیدا کردن بهترین مکانیسم ارتباطی
  - مشخص کردن اهداف کلی و دامنه ی آن اهداف
  - توجه به جزئیات
    - مشتریان باید یک سناریو تهیه کنند.
    - در آوردن نقاط مهم سناریو را از درون آن.
  - بازبینی نتایج به همراه مشتریان و کاربران نهایی(ذی نفع ها)

Practice های برنامه ریزی

- شامل مجموعه ای از Practiceهای تکنیکی و مدیریتی است که تیم نرم افزاری را قادر به تعیین یک نقشه ی راه با استفاده از اهداف تکنیکی و استراتژیکی می کند.
- اصول
  - فهمیدن هدف پروژه
  - درگیر شدن (بحث و مناظره) با مشتریان (و سایر ذی نفع ها)

Translated by: mohammad madvari

- درک اینکه برنامه ریزی یک عمل تکراری و بازگشتی است.
- تخمین زدن چیزهایی که نیاز دارین
- در نظر گرفتن ریسک ها
- واقع گرا باش
- سطح جزئیات پروژه را مشخص کنیم.
- تعریف اینکه چگونه به کیفیت مورد نظر برسیم.
- تعریف اینکه اگر تغییراتی رخ داد چگونه خود را با آنها وفق دهیم.
- بررسی مجدد برنامه و برقرار کردن تعادل بین محتویات برنامه

Practice های برنامه ریزی

- برای توسعه ی یک برنامه واقعی برای پروژه به چه سولاتی باید پاسخ داده شود؟
  - سولات W5HH بوهم در این زمینه عبارتند از:
    - چرا این سیستم باید تولید شود؟
    - چه چیزی رو می خواهد انجام دهد؟
    - چه زمانی کار را انجام خواهد داد؟
    - چه کسی مسئول انجام عملیات است؟
    - این سیستم (پروژه) برای کجا (چه نهادی) طراحی می شود؟
    - از لحاظ تکنیکی و مدیریتی چگونه کارها را انجام می دهد؟
    - به چه میزان از هر یک از منابع نیاز است؟

Practice های برنامه ریزی

- یک مجموعه کاری خلاصه شده
  - هدف پروژه را دوباره بررسی کن.
  - ریسک های پروژه را ارزیابی کن.
  - بررسی توابع و خصوصیات
  - ساختار توابع و خصوصیت ها را بررسی کن.
  - ایجاد یک برنامه کلی
    - تعداد increment های نرم افزار
    - زمان بندی کلی
    - تاریخ تحویل هر کدام از increment ها
  - مشخص کردن یک ریزبرنامه برای increment اول
  - بررسی پیشرفت

Practice های مدل سازی

- مشخص کردن مدلهایی که ما را در تولید موجودیت اصلی کمک می کنند.شامل دو قسمت زیر

Translated by: mohammad madvari

است :

- ***مدل های تحلیلی : که نیازمندی های کاربر را در سه دامنه مشخص می کنند: دامنه اطلاعات.دامنه تابعی.دامنه رفتاری***
- ***مدل های طراحی : که خصوصیات نرم افزار را به ما نشان می دهند.این خصوصیات به پیاده کنندگان سیستم (برنامه نویسان) کمک شایانی می کند.و شامل مدل طراحی معماری . مدل طراحی واسط کاربری و مدل طراحی سطح کامپوننت است.***

Practice های مدل تحلیل

- **مدل تحلیل بر ۳ ویژگی نرم افزار تمرکز دارد:**
  - **اطلاعاتی که باید پردازش شوند.**
  - **عملکردی که باید انجام شود.**
  - **رفتاری که باید نمایش داده شود.**
- **اصول مدل تحلیل عبارتند از:**
  - **ارائه و فهم دامنه ی اطلاعات مساله**
  - **تعریف توابعی که نرم افزار آن ها را اجرا می کند.**
  - **نمایش رفتارهای نرم افزار در ارتباط با رخدادهای بیرونی**
  - **دسته بندی موارد اطلاعاتی . عملیاتی و رفتاری**
  - **شروع کار پیاده سازی با استفاده از دسته بندی های انجام شده**
- **اجزای مدل تحلیل عبارتند از :**
  - **مدل داده ای**
  - **مدل جریانی**
  - **مدل کلاسی**
  - **مدل رفتاری**

Practice های مدل طراحی

- **اصول**
  - **طراحی باید در مدل تحلیل قابل ردگیری باشد.**
  - **همیشه معماری را در نظر داشته باشد.**
  - **بر روی طراحی داده ها متمرکز شو.**
  - **واسط ها(درونی و بیرونی) باید با دقت طراحی شوند.**
  - **طراحی واسط کاربری را با نیاز های کاربر نهایی وفق بده.**
  - **اجزا باید استقلال توابع را نشان بدهند.**
  - **اجزا باید ارتباط ضعیف (loosely coupled) باشند.**
  - **ارائه ی طراحی باسد قابل فهم باشد.**
  - **مدل طراحی باسد به طور بازگشتی توسعه داده شود.**
- **اجزای مدل طراحی**
  - **طراحی داده ای**

Translated by: mohammad madvari

- طراحی معماری
- طراحی اجزا
- طراحی واسط کاربری

Practice های ساخت

- **اصول مقدماتی: قبل از اینکه حتی یک خط کد بنویسید مطمئن شوید که :**
  - مساله ای را که می خواهید حل کنید به خوبی فهمیده اید.
    - از ایجاد یک برنامه ی زیبا ولی با روش حل نادرست اجتناب کنید.
  - اصول و مفاهیم اولیه ی طراحی را بفهمید.
  - یک زبان برنامه نویسی که نیازهای برنامه را برطرف می کند انتخاب کنید.
  - محیطی را برای برنامه نویسی انتخاب کنید که ابزارهای آن کار شما را ساده تر می کنند.
  - مجموعه ای از Unit test هایی را بسازید که وقتی کد شما کامل شد بتواند بر روی آن اعمال شود.

Practice های ساخت

- ***اصول کد نویسی: وقتی که خواستید کدنویسی را شروع کنید به این نکات دقت کنید:***
  - الگوریتم های برنامه تان را با استفاده از practice های برنامه نویسی ساخت یافته محدود کنید.
  - ساختمان داده ای را که نیازهای طراحی شما را برطرف می کند انتخاب کنید.
  - معماری نرم افزار را بفهمید و واسط کاربری سازگار با آن را ایجاد کنید.
  - سعی کنید از منطق های شرطی استاده کنید چون نگه داری کد را آسان تر می کنند.
  - حلقه های تودرتو را طوری بنویسید که امتحان کردن آنها ساده باشد.
  - از نام های متغیر با معنی استفاده کنید و همچنین از سایر استانداردهای کدنویسی تبعیت کنید.
  - سعی کنید کدتان خود مستند باشد.
  - یک نمای ظاهری (حاشیه گذاری . خطوط فاصله و ...) که به فهم بهتر کمک می کند را طراحی کنید.

Practice های ساخت

- ***اصول اعتبار سنجی: بعد از اینکه اولین مرحله از کدنویسی را انجام دادید دقت کنید که:***
  - در یک فرصت مناسب جلسه ای را برای بررسی کد تشکیل دهید.
  - Unit test ها را اجرا کنید و خطاهای کشف شده را اصلاح کنید.
  - عملیات Refactor را بر روی کد اعمال کنید.

Practice های ساخت

- **اهداف امتحان نرم افزار چیست؟**
  - پیدا کردن خطا ها (حتی یک خطا)

Translated by: mohammad madvari

- یک تست خوب به احتمال زیاد خطاهایی که تاکنون کسف نشده اند را پیدا می کند.
- یک تست موفق خطاهای تاکنون کشف نشده را پیدا می کند.

- اصول تست
  - همه ی تست ها باید برای نیازمندی ها قابل ردیابی باشند.
    - تست های عملکردی
  - تست ها باید قبل از شروع خوب برنامه ریزی شوند.
  - اصول Pareto باید به تست اعمال شوند.
    - ۸۰ درصد از خطاها می توانند در ۲۰ درصد از کد پیدا شوند.
  - تست از کم شروع می شود و رفته رفته زیاد می شود.
  - تست جامع و کامل امکان پذیر نیست.

Practice های توسعه

اصول

- مدیریت انتظارات مشتری در هر increment
  - مطمئن شوید از اینکه مشتری در هر مرحله از پیشرفت بداند چه چیزهایی را بدست می آورد.
- یک Package برای تحویل باید پیاده سازی و امتحان شود.
- یک روش حمایتی (از محصول) باید پیاده سازی شود.
  - کاربران نهایی باید یک شخص رابط داشته باشند.
- دستورالعمل های کاری باید برای کاربران نهایی فراهم شوند.
- نرم افزارهای مشکل دار ابتدا باید مشکلشان حل شود و سپس تحویل داده شوند.
  - مشتریان از اینکه یک محصول با کیفیت بالا را در مدت زمان کمی به آنها تحویل داده اید را فراموش می کنند ولی هیچ وقت فراموش نمی کنند که مسائل و مشکلات باعث تحویل یک محصول با کیفیت پایین می شود.

بازخورد

- **بازخورد مشتری راهنمایی برای کسب رضایت مشتری است.**
  - ویژگی ها . عملکرد ها . سهولت استفاده . قابلیت اطمینان و ...
- **از بازخورد استفاده می شود برای :**
  - برای اصلاح سریع برنامه در صورت نیاز
  - برای مشخص کردن تغییرات

Translated by: mohammad madvari

- برای اصلاح طراحی
- اصلاح برنامه برای بازتاب تغییرات

خلاصه

- مجموعه ای از اصول مدیریتی و تکنیکی ضروری هستند اگر بخواهیم Practice های مهندسی نرم افزار خوبی داشته باشیم.
  - اصول تکنیکی شامل اصولی برای فهم نیازمندی ها و ناحیه های نمونه اولیه ی نامشخص و نیاز به تعریف واضح معماری نرم افزار و برنامه ریزی برای کامل کردن اجزا می شود.
  - اصول مدیریتی شامل اصولی برای تعریف اولویت ها و زمان بندی واقعی و نیاز به مدیریت ریسک و سنجش کیفیت می باشد.

فصل ۶

مهندسی سیستم

مهندس سیستم

- مهندسی نرم افزار بعنوان نتیجه ای از مهندسی سیستم تلقی می شود.
- تمرکز مهندسی سیستم بر سازمان دهی اجزای یک سیستم است که این اجزا می تواند یک محصول یک سرویس یا یک تکنولوژی برای تبدیل اطلاعات یا کنترل باشد.
- فرایند های مهندسی سیستم در شکل های گوناگونی وابسته به حوزه ی کاربرد آن ظاهر می شوند.
  - مهندسی فرایند تجاری یک زمینه ی کاری قابل توجه در سازمان تجاری است.
  - وقتی یک محصول می خواهد ساخته شود فرایند بنام مهندسی محصول شناخته می شود.

Translated by: mohammad madvari

مهندسی سیستم چیست؟

- تعیین می کند:
  - هدف کلی سیستم
  - نقش سخت افزار نرم افزار افراد پایگاه داده روال ها و سایر اجزای سیستم
  - نیازمندی های عملیاتی باید
    - تعیین شده باشند
    - تحلیل شده باشند
    - مشخص شده باشند
    - مدلسازی شده باشند
    - اعتبار سنجی شده باشند
    - مدیریت شده باشند
- برای انجام این کار مهندس سیستم با مشتریان و کاربران نهایی و سایر ذی نفع ها(برای فهم بهتر نیازمندی ها)همکاری می کند.

محصول کاری عمده

- یک ارائه ی موثر از سیستم باید بعنوان نتیجه ی مهندسی سیستم تولید شود.
  - نمونه اولیه
  - تشخیص
  - مدل سمبولیک
- هر چیزی که انتخاب بشه باید با ویژگی های سیستمی زیر ارتباط برقرار کند.
  - عملیاتی بودن
  - عملکرد
  - رفتار
- ارائه باید بینشی را نسبت به معماری سیستم فراهم کند.

سیستم کامپیوتری

- تعریف :
  - مجموعه ای منظم از اجزا که برای انجام دادن بعضی اهداف از قبل تعریف شده (با استفاده از پردازش اطلاعات) با هم سازمان دهی شده اند.

مهندسی سیستم

- اجزای یک سیستم کامپیوتری
  - نرم افزار

Translated by: mohammad madvari

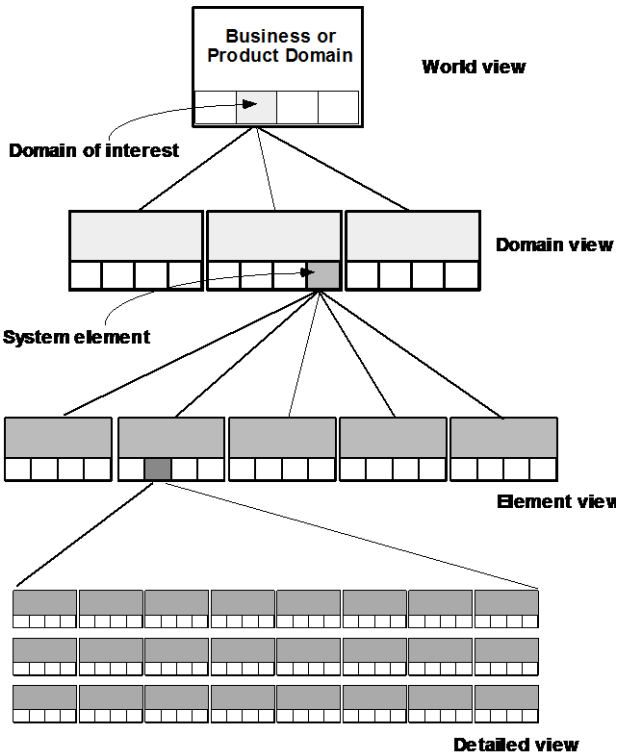
- سخت افزار
- افراد
- پایگاه داده
- مستندات
- روال ها

- توجه:این فقط نرم افزار نیست.
- یک سیستم پیچیده سلسله مراتبی از اجزای بزرگ است.
  - نقش مهندس سیستم تعریف اجزای کلی سلسله مراتب سیستم های منفرد و یا اجزای بزرگ است.

سلسله مراتب مهندسی سیستم

- مجموعه روش های بالا به پایین و پایین به بالا برای کنترل سلسله مراتب
  - در بالای سلسله مراتب یک زمینه و موضوع گسترده قرار دارد.
  - در پایین سلسله مراتب فعالیت های تکنیکی با جزئیات توسط مهندسین مربوطه انجام می شود.
    مثل فعالیت های سخت افزاری یا نرم افزاری
- مهندسی سیستم خوب با یک فهم درست از زمینه ی کاری (world view) کارش را شروع می کند و سپس تمرکز خود را به جزئیات تکنیکی قابل فهم محدود می کند.

سلسله مراتب مهندسی سیستم



Translated by: mohammad madvari

مدلسازی سیستم

- هدف مدل مهندسی سیستم چیست؟
  - فرایند هایی را برای برآورده کردن نیازهای view ی مورد نظر تعریف می کند.
  - رفتار فرایندها و فرض های مربوط به آن ها را نشان می دهد.
  - به طور واضح ورودی های درونی و بریونی مدل را تعریف می کند.
    - ورودی های بیرونی یک جزء از viewی موجود را با سایر اجزای هم سطح در سطوح دیگر ربط می دهند.
    - ورودی های درونی اجزای یک جزء بزرگتر را در هر viewی خاص به هم مرتبط می سازند.
  - تمام پیوند هایی که مهندس را به فهم بهتر View قادر می سازد نشان می دهد.

ایجاد مدل سیستمی

- مهندس سیستم وقتی راه حل های پیشنهادی را بررسی می کند فاکتورهای زیر را در نظر می گیرد:
  - فرض ها: برای کاهش تعداد تبدیل ها و دگرگونی ها
  - ساده سازی: برای اینکه مدل را قادر به ایجاد با یک روش زمان دار نماید.
  - محدودیت ها : برای کمک به محدود کردن سیستم
  - اجبارها: برای هدایت روش هنگامی که مدل ساخته می شود و برای استفاده آماده است.
  - برتری ها: معیاری برای شناسایی معماری برتر برای تمام داده ها توابع و تکنولوژی
- مدل ما چقدر مفید خواهد بود؟
  - سیستم واکنشی – سیستم ماشین های کنترل فرایند ها و افراد را بکار می گیرد.
    - مخصوصا برای یک سیستم واکنشی اگر قابلیت شبیه سازی وجود نداشته باشد ریسک پروژه بالا می رود و آن وقت باید از فرایند increment استفاده شود.

مدل فرایند تجاری (BPE)

معماری های سیستم

- سه نوع معماری متفاوت در زمینه ی اهداف تجاری باید تحلیل و طراحی شود:
  - معماری داده ای (data architecture)
  - معماری کاربردها (applications architecture)
  - زیرساخت فناوری (technology infrastructure)
- *معماری داده ای یک چارچوب برای نیازهای اطلاعاتی یک Business یا یک Business function فراهم می کند.*
- *معماری کاربردی اجزایی را شامل می شود که داده های ما را به چیزی که هدف ما آن را دنبال می کند تبدیل می نماید.*
- *زیر ساختار فناوری شالوده ی اصلی برای معماری داده ای و کاربردی را فراهم می کند.*

Translated by: mohammad madvari

سلسله مراتب BPE

- برنامه ریزی استاتژیک اطلاعات (ISP)
  - اهداف استراتژیک مشخص می شود.
  - فاکتورهای موفقیت شناسایی می شود.
  - مدل سازمانی طراحی می شود.
- تحلیل ناحیه تجاری (BAA)
  - پروسه ها و سرویس های مورد نیاز مدلسازی می شود.
  - ارتباط پروسه ها و داده ها مشخص می شود.
- مهندسی Application
  - مهندسی نرم افزار صورت می گیرد.
  - بر اساس محدودیت های سازمان و اهداف آن مهندسی صورت می گیرد.
- ایجاد و تحویل
  - استفاده از روشهای تست CASE و 4GTs

برنامه ریزی استراتژیک اطلاعات (ISP)

موضوعات مدیریتی

- تعریف اهداف تجاری استراتژیک
- نگهداری از فاکتورهای موفقیت حساس و حیاتی
- هدایت کردن تحلیل تکنولوژیک
- انجام تحلیل برای سیستم های استراتژیک

موضوعات تکنیکی

- ایجاد یک مدل داده ای سطح بالا
- دسته بندی کردن بر اساس ناحیه تجاری/سازمانی
- تصحیح مدل و دسته بندی

تعریف اهداف و جهت گیری ها

- Objective – مشخص کردن جهت حرکت (جهت گیری سازمان)

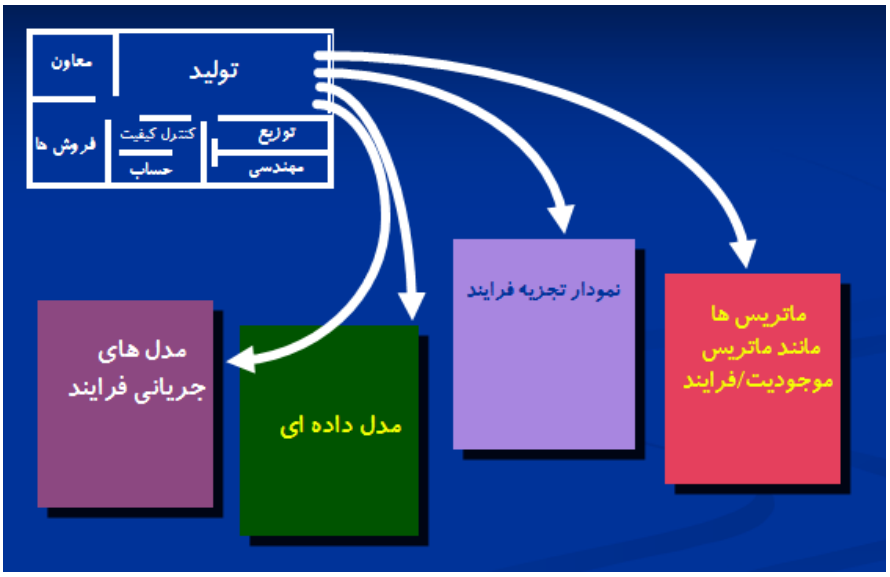
Translated by: mohammad madvari

- objective – Goal های قابل اندازه گیری را تعریف می کند.
  - زیراهداف (Subgoals)
    - نرخ reject را در ۶ ماهه اول ۲۰ درصد کاهش می دهد.
    - ۱۰ درصد قیمت امتیاز تولیدکنندگان را بدست می آورد.
    - ۳۰ درصد اجزا را برای سهولت ساخت در سال اول باز مهندسی (re-engineer)می کند.
- Objective ها استراتژی اند در حالی که اهداف (goals) عملی اند.

تحلیل محیط تجاری (BAA)

- گروه هایی طبیعی را برای تولید function و data تعریف می کند.
- تعدادی از فعالیت های آن مانند ISP است با این تفاوت که این کارها را با دیدی محدودتر انجام می دهد.
- سیستم های اطلاعاتی موجود را شناسایی می کند تا تعیین کند که با مدل های جدید ISP سازگار هستند یا خیر.
  - سیستم هایی که با مشکل روبرو می شوند را شناسایی می کند.
  - سیستم هایی که با مدل های اطلاعاتی جدید ناسازگارند را مشخص می کند.
  - پس از اولویت بندی کارها شروع به مهندسی مجدد می کند.

فرایند BAA



Translated by: mohammad madvari

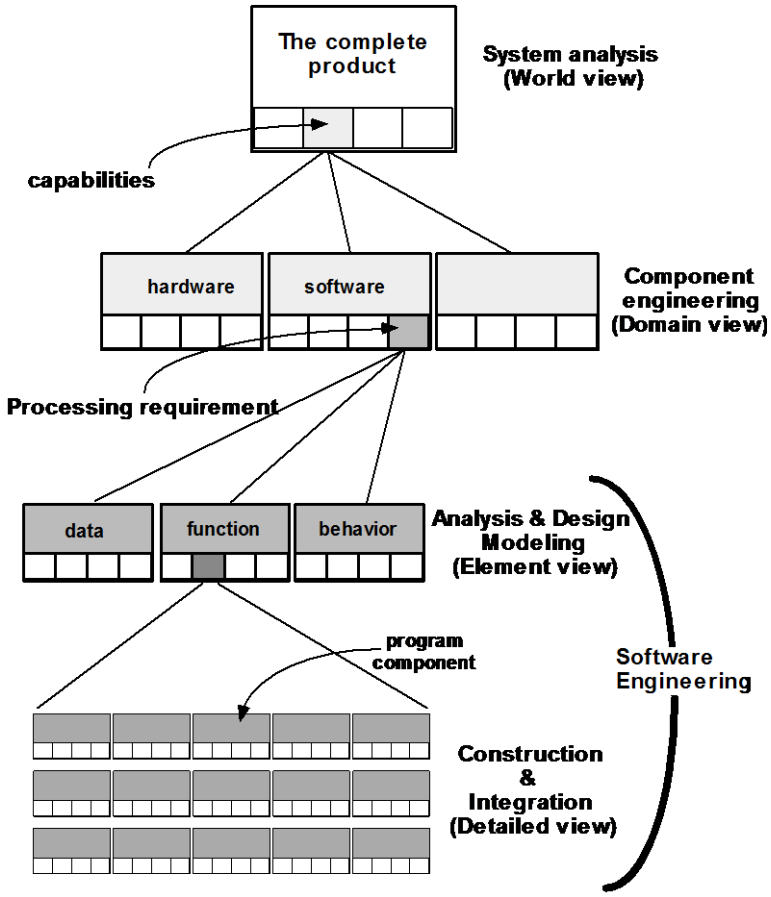
مهندسی محصول

- هدف آن برآورده کردن خواسته های مشتری برای مجموعه ای از قابلیت های یک محصول است.
- مانند BPE شامل قسمت هایی است که عبارتند از:
  - اجزای معماری
    - نرم افزار
    - سخت افزار
    - پایگاه داده
    - افراد
  - زیر ساختار
    - فناوری
- دید جهانی (world view)از طریق مهندسی نیاز مندی ها بدست می آید.
  - نیازمندی های کلی مشتری جمع آوری می شوند.
    - نیاز های اطلاعاتی و کنترلی
    - عملکرد و رفتار محصول
    - کارایی
    - محدودیت های رابط کاربری و طراحی
  - مهندسی نیاز ها عملکرد و رفتاری را به هر کدام از ۴ جزء بالا تخصیص می دهد.

- **وقتی که تخصیص انجام شد مهندسی اجزای سیستم شروع می شود.**
  - اغلب از مدل فرایندی همزمان (CPM) استفاده می شود.
  - فرایند مهندسی روی هر کدام از اجزا (نرم افزار–سخت افزار–پایگاه داده–افراد) به صورت موازی صورت می گیرد.
- **دید عناصر (Element view) به صورت خود مهندسی است.**
  - برای مهندسی نرم افزار این به معنی مدلسازی تحلیل و طراحی و ساخت و تولید و تست و پشتیبانی است.

Translated by: mohammad madvari

مهندسی محصول



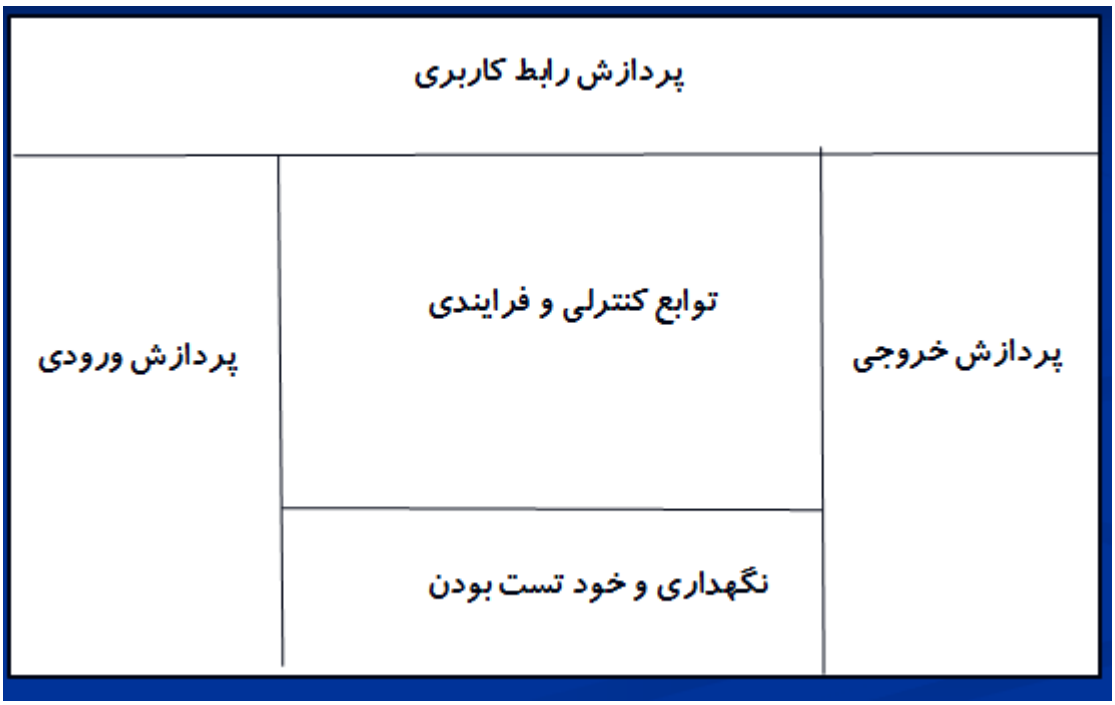
مدلسازی سیستم

- یک سیستم در سطوح مختلفی می تواند ارائه شود (world-domain-element)
- مدلهای سیستمی برای اینکه سلسله مراتب را در حالت طبیعی خود نگه دارند از آن مراقبت می کنند.

Translated by: mohammad madvari

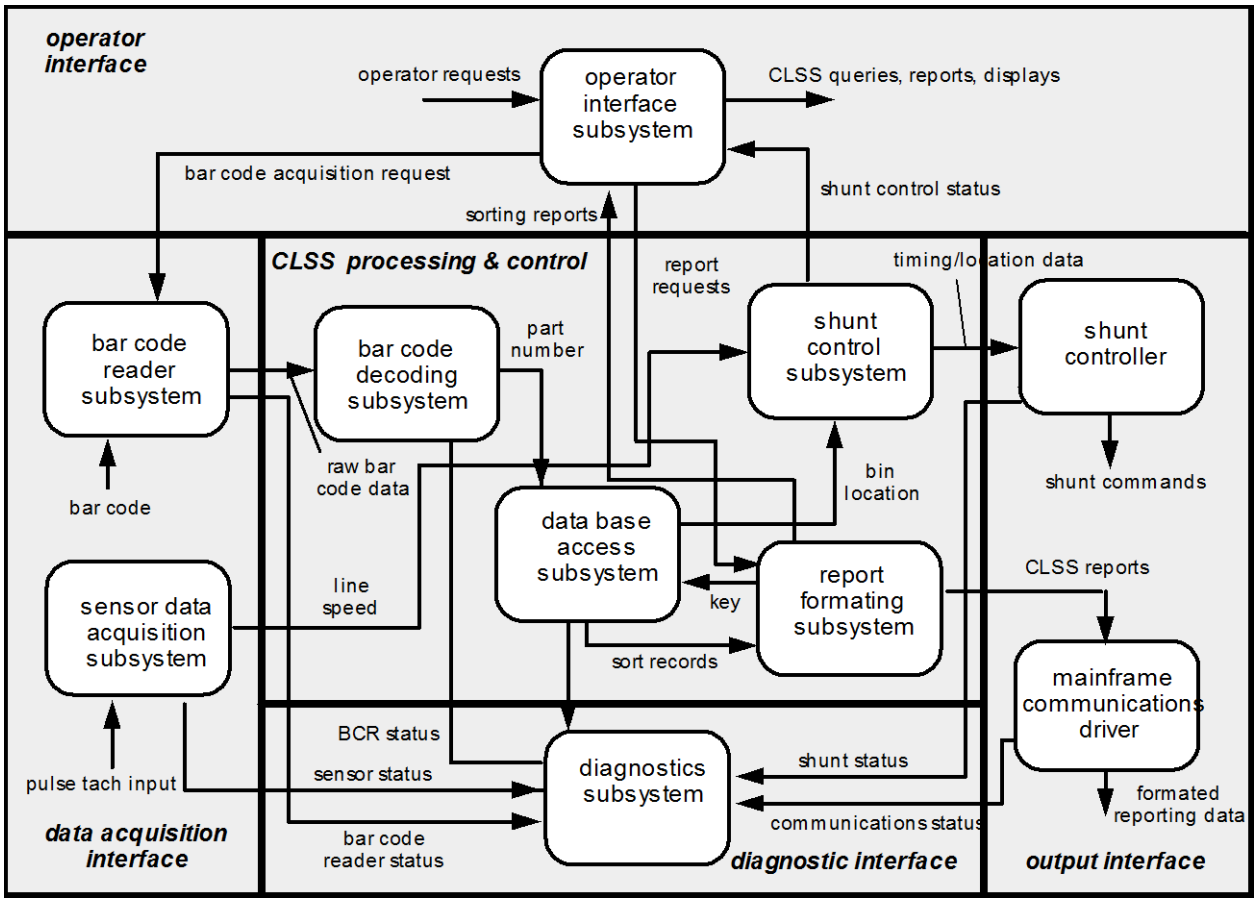
- در اینجا ما ۲ مدل را مشاهده می کنیم:
  - Hately-Pribhari و UML
- سیستم های کامپیوتری مدل H-P از یک الگوی ورودی-پردازش-خروجی به همراه رابط کاربری و نگهداری و تست استفاده می کنند.
  - مثل سیستم مرتب سازی نوار نقاله (CLSS)

الگوی معماری محصول



Translated by: mohammad madvari

نمودار جریانی معماری



مدلسازی سیستمی با استفاده از UML

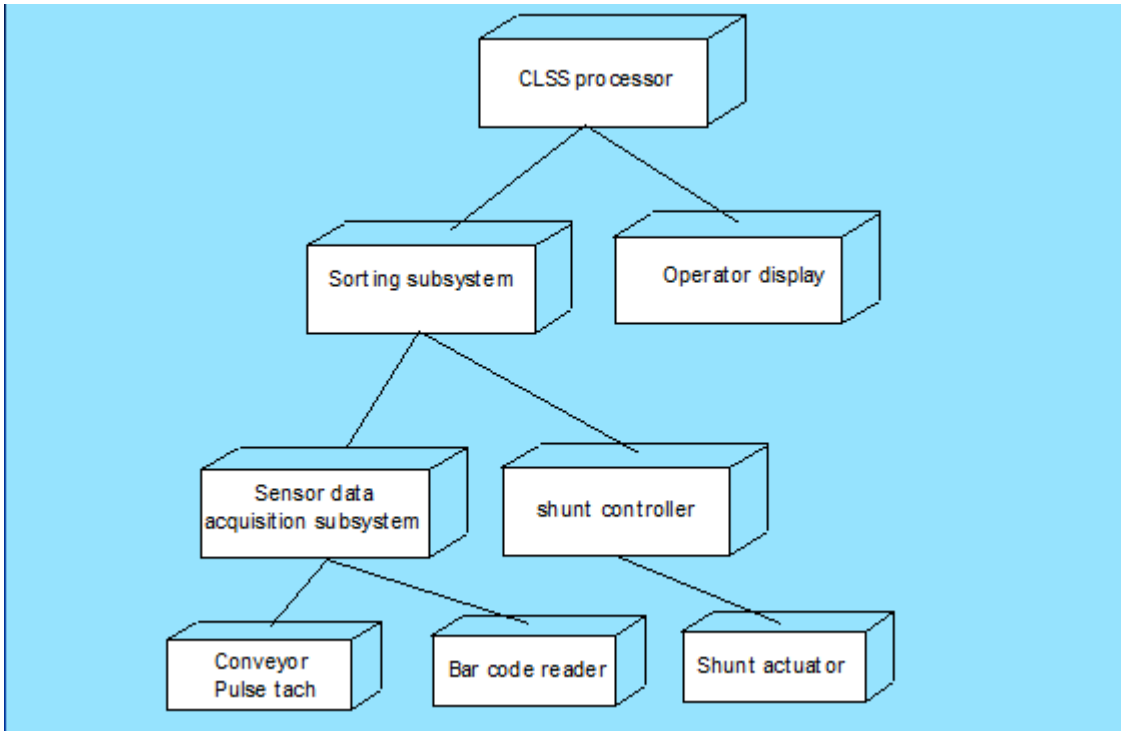
- UML نمودارهای متنوعی را برای تحلیل و طراحی سطوح سیستمی و نرم افزاری فراهم می کند.

Translated by: mohammad madvari

- در اینجا ما به مدلسازی اجزای ۴ گانه سیستمی نیاز داریم.

- **نمودارهای توسعه**
  - هر جعبه ی ۳بعدی یک جزء سخت افزاری را نشان می دهد که خود جزئی از معماری فیزیکی سیستم است.

نمودار توسعه

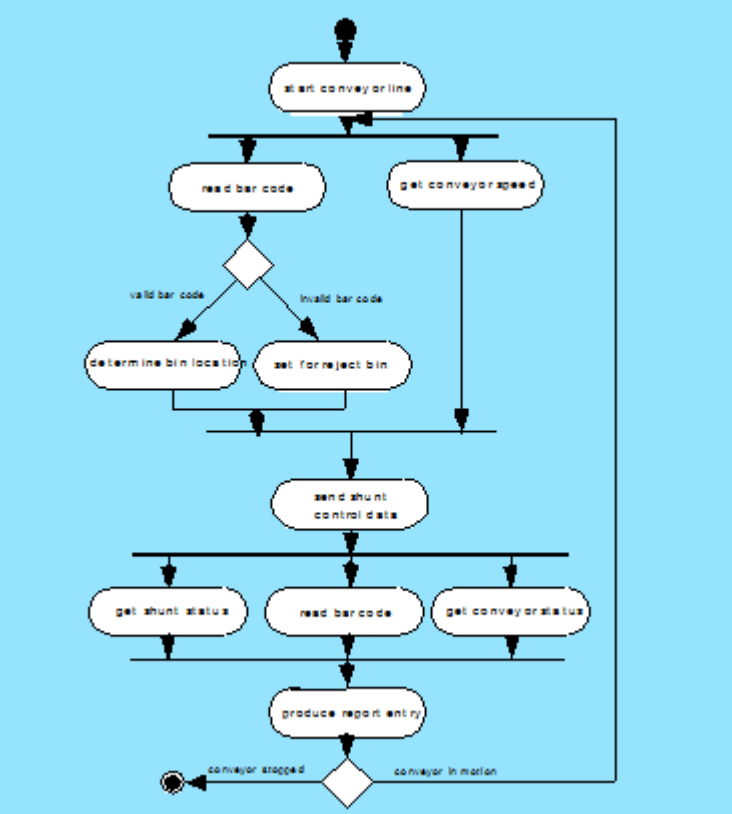


مدلسازی سیستمی با استفاده از UML

- **نمودارهای فعالیت**
  - جنبه های روالی یک جزء سیستمی را نشان می دهد.

Translated by: mohammad madvari

نمودار فعالیت

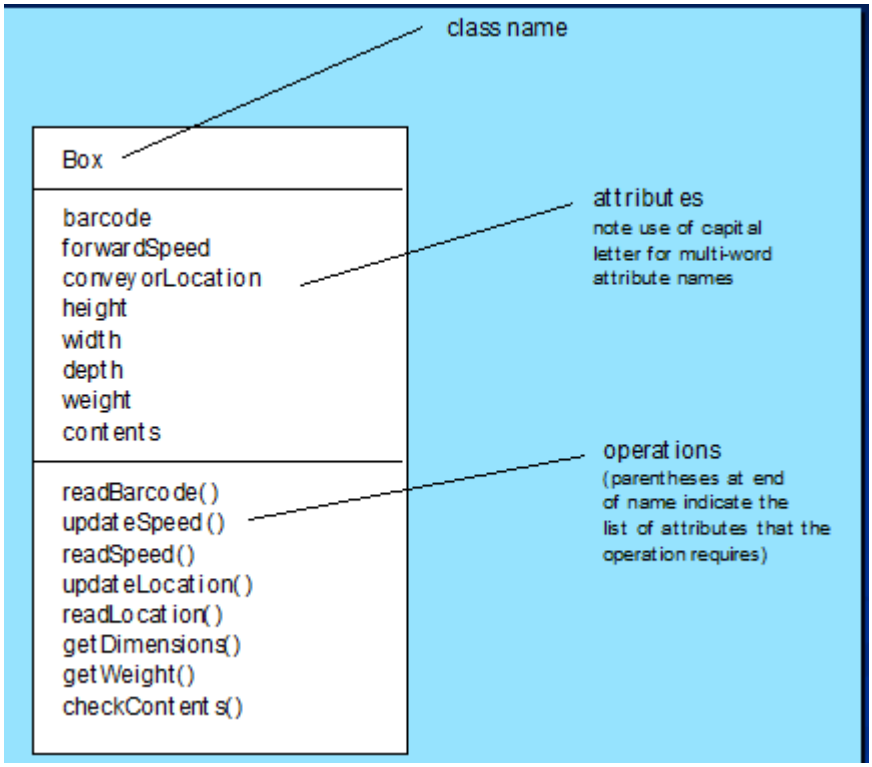


مدلسازی سیستمی با استفاده از UML

- **نمودارهای کلاسی**
  - **اجزای سیستم را در قالب داده هایی به همراه توصیف ویژگی ها و عملکردهای آن ها نشان می دهد.**
- **عملیات می تواند با استفاده از یک نمودار Use-Case مدلسازی شود.**

Translated by: mohammad madvari

نمودار کلاسی



Translated by: mohammad madvari

خلاصه

- مهندسی سیستم کمک می کند تا نیاز های مشتری در قالب یک مدل برآورده شوند.
- با یک دید جهانی (world view) شروع می شود.
- وقتی که هر کدام از اجزای سیستم تحلیل شد در یک سطح دیگر بنام دامنه به آن متمرکز می شوند.
- BPE رویکردی از مهندسی سیستم است که برای تعریف معماری هایی که یک Business را قادر به استفاده ی موثر از اطلاعات می کند استفاده می شود.
- مهندسی محصول یک رویکرد از مهندسی سیستم است که برای مشخص کردن نیازهای مشتری . تعیین امکانات اقتصادی و فنی و تخصیص عملیات و کارایی به نرم افزار .سخت افزار.افراد و پایگاه داده استفاده می شود.